REMARKS

Claims 1-123 are pending in this application. Claims 1-18, 29-41, 45-59, 70-82, 86-100 and 111-123 have been rejected. By this paper, the Applicant amends claims 83-123 for clarification. No new matter has been added.

At page 2 of the Office Action, the Examiner rejects claims 1-123 under 35 USC § 101 as claiming no more than an abstract idea. The Applicant traverses this rejection. The Applicant respectfully disagrees that the rejected claims recite no more than an abstract idea. The Examiner states that the rejected claims are non-statutory "because they do not disclose that how a method will carry out its functionality and perform native binding to execute native code with incorporating further steps of translating and processing." Regarding independent claim 1, this claim concerns "A method of performing native binding to execute native code during the translation of....". Claim 1 therefore clearly relates to a patentable process in relation to a physical apparatus – i.e., a subject processor – which performs the claimed execution of native code. The Applicant disagrees with the Examiner's contention that claim 1 is unable to produce useful results and derive a practical application. The Applicant submits that there is a clear, useful and practical result of the claimed process, namely that translation of subject program code into target code is achieved more efficiently by a computer when performing the claimed method. The Applicant therefore believes claim 1 is statutory under 35 U.S.C § 101, so this rejection is improper and should be withdrawn. Claims 2-41 depend from allowable claim 1, and so those claims should also be allowable.

Regarding independent claim 42, that claim is directed to "A computer-readable storage medium having software resident thereon in the form of computer-readable code executable by a computer to perform. Claim 42 is a method equivalent of claim 1, and so is directed to statutory subject matter for the same reasons presented above for claim 1. Contrary to the Examiner's suggestion, the Applicant respectfully submits that independent claim 42 does not define intangible embodiments. Claim 42 is directed to a computer-readable storage medium having software resident thereon in the form of computer-readable code executable by a computer to perform. Accordingly claim 42 should be allowable. The rejection is improper and should be

26

withdrawn. Claims 43-82 depend from allowable claim 42, and so those claims should also be allowable.

Claim 83 is directed to a combination of a target processor and "translator code for performing native binding <u>to execute native code</u> during the translation of . . ." Therefore, as set forth above for claim 1, the Applicant believes claim 83 is directed to a patentable process in relation to a physical apparatus, and that there is a clear, useful and practical result – more efficient translation of subject program code into target code. The Applicant respectfully believes that this rejection is improper and should be withdrawn. Claims 84-123 depend from allowable claim 83, so those claims should also be allowable.

At page 3 of the Office Action, the Examiner rejects claims 1-123 under 35 U.S.C. §112 as being incomplete for omitting essential elements, namely "instructions, properly compilation, assembling, loading and linking." The Applicant traverses this rejection. The Applicant respectfully submits that the elements to which the Examiner refers have nothing to do with the claimed invention. The claimed invention clearly does not relate to compilation or assembly, and so the Examiner's suggestions regarding omitted elements of "instructions, properly compilation, assembling, loading and linking" are unnecessary and improper. The Applicant respectfully submits that the independent claims 1, 42 & 83 include all of the essential elements of the invention, and no gaps exist between those elements. The Applicant notes that the cited prior art of Adams (US 6,578,193) describes steps which include such instructions, compilation, assembly, loading and linking but, as will become clear from the detailed comments below, features which might indeed be essential in the system of Adams are certainly not essential in the invention of claims 1, 42 and 83. The Applicant submits that these rejections are improper and should be withdrawn.

The Examiner rejects claims 83-123 for use of the introductory portion "In combination." Although the Applicant believes that the original claim language is clear and proper, to expedite prosecution the Applicant amends claim 83 to remove the phrase "in combination," and to include the phrase "A computing apparatus." Accordingly, that rejection should be withdrawn and claim 83 as amended should be allowable. Claims 84-123 have been amended to include language corresponding to claim 83, so those claims as amended should also be allowable.

27

At page 4, the Examiner rejects claims 1-18, 29-41, 45-59, 70-82, 86-100 and 111-123 under 35 U.S.C. §103 as being unpatentable over U.S. Patent No. 6,578,193 (Adams) in view of U.S. Patent No. 6,578,194 (Baumgart).

The Adams patent (US 6,578,193)

Adams discloses an endian-neutral loader for an interpretive environment. Column 1, lines 9-58 of Adams explains the field of that interpretive environment relating to an accelerator for emulators and interpreters for <u>virtual machine environments</u> such as Java or Visual Basic. We trust that the Examiner is familiar with this form of virtual machine environment, whereby platform-independent Java bytecodes are executed on a particular hardware host machine through an interpreter or emulator program (the "abstract machine" or "virtual machine"). The Examination Report refers particularly to Figures 3, 11 & 12 and the description at column 8, lines 44-51 concerning "the builder 86" of Adams. Here, Adams describes the builder 86 which <u>builds the virtual machine</u> by creating an executable 100 in the main memory 18, 20 associated with the processor 12 of the host hardware.

Adams is particularly concerned with the problem of lack of performance achieved by interpretive programs such as Java bytecodes because of the mismatch between byte-ordering of the virtual machine and the <u>host hardware</u>. Thus, Adams discloses a loading technique which mitigates this byte ordering problems.

However, from this proper understanding of Adams, the Applicant submits that Adams does not teach or suggest the limitations of independent claims 1, 42 or 83.

For example, claim 1 requires "[a] method of performing native binding to execute native code during the translation of subject program code executable by a subject processor to target program code executable by a target processor, wherein the native code is code executable by the target processor". Adams simply does not translate subject program code executable by a subject processor into target program code executable by a target processor, as required by this limitation of claim 1.

Further, Adams does not perform the step of "identifying certain subject program code [executable by a subject processor] having corresponding native code [executable by the target processor]" as in claim 1. Here, the Examiner refers to the builder 86 of column 8, lines 44-51 of Adams. The builder 86 does not perform the identifying step of claim 1. Instead, as noted above, the builder 86 builds the executable 100 to provide the virtual machine environment on the host hardware.

It follows that Adams cannot disclose the step of "identifying the native code which corresponds to the identified subject program code" in claim 1. Here the Examiner refers to column 8, lines 52-65 which does indeed mention "native code". However, it is also clear that the "native code" mentioned in Adams refers to the code which is executed directly by the processor 12 of the host hardware containing processor-specific instructions, i.e., instructions which are specific to a particular type of host hardware and in contrast to the platform-independent virtual machine instructions (Java bytecodes). This "native code" in Adams does not "correspond to the identified subject program code" as in claim 1.

We agree with the Examiner that Adams does not teach explicitly "executing the corresponding native code instead of executing a translated version of the identified subject program code" as in the third step of claim 1.

Here, it is helpful to look again at what the present invention is trying to achieve. Paragraphs [0008] and [0009] of the description provide a useful summary of the invention:

> In particular, the inventors have developed an optimization technique directed at expediting program code conversion, particularly useful in connection with a run-time translator which employs translation of subject program code into target code. A native binding technique is provided for inserting calls to native functions during translation of subject code to target code, such that function calls in the subject program to subject code functions are replaced in target code with calls to native equivalents of the same functions. Parameters of native function calls are transformed from target code representations to be consistent with native code representations, native code calling conventions, and native function prototypes.
>
> Native binding is the mechanism of the present invention that enables translated subject code to execute a native (i.e., target architecture) version of a unit of code directly, rather than translating and executing the equivalent subject code, such as, for instance, a native version of a library. This avoids the overhead of translating the subject versions of those libraries. In addition, the native version may be a much more efficient implementation of the same

functionality, as the native version can exploit architectural features of the target architecture which the subject version is unaware of. (emphasis added).

Thus, the whole point of the present claims is that instead of translating an entire subject program into a translated target program executable on the target processor, there are certain portions of the subject program (such as function calls to proprietary libraries or system libraries) which can instead be executed by identifying and executing an already existing native code version. This "native binding" mechanism saves the work of having to translate from subject code to target code - with respect to those identified portions of subject code. Also, this "native binding" mechanism allows the translation process to make use of native code which is written specifically for the host (target) hardware and is thus better able to exploit architectural features of the host hardware than the target code.

The Baumgart patent (US 6,578,194)

In simple terms, Baumgart discloses a method to enhance assembly or compilation from a high-level human-oriented programming language such as COBOL or FORTRAN down into machine-readable object code. In contrast, the present claims are directed to an improvement in translation from one machine-readable subject program (the subject program code executable by a subject processor) into machine-readable executable code executable by a different type of processor (the target code executable by the target processor). In other words, Baumgart concerns human to machine translation. The present invention concerns machine to machine translation.

Baumgart does not supply that which is missing from Adams. Baumgart concerns an assembler or compiler which converts a source program written in a high-level programming language such as FORTRAN or COBOL into machine-readable executable code. (See column 1, lines 15-27). The Examiner specifically refers to column 2, lines 30-44 "the computer translates the source program [i.e. a high-level source language such as FORTRAN or COBOL] into an object module [i.e. a machine readable executable code]...". Clearly, this "translation" from source code to object code in Baumgart is irrelevant to the claimed method of translating from subject code executable by a subject processor into target code executable by a target processor as in the claimed invention.

30

Further still, it is unnecessary and impractical to combine the human-to-machine translation system of Baumgart with the platform-independent virtual machine environment of Adams. These are two entirely separate approaches to the process of producing code which is executable by a particular host hardware. In particular, it would not have been obvious to a person of ordinary skill in the art to combine Adams with Baumgart because such a combination is unnecessary and impracticable. Even so, the combination of teachings from Adams and Baumgart still does not arrive at the claimed invention. For at least the reasons set forth above, claim 1 should be allowable. Claims 2-41 depend from allowable claim 1, so those claims should also be allowable.

The arguments set forth above also apply to independent claims 42 & 83. Accordingly, claims 42 and 83 should be allowable. Claims 43-82 depend from allowable claim 42, and claims 84-123 depend from allowable claim 83, so those claims should also be allowable.

Further, the dependent claims each relate to additional non-obvious features. For example, referring to the dependent claims 2, 43 & 84 Baumgart does not teach the recited limitation of "executing the native function instead of the subject function in the translation of the subject program code" as alleged in the Examination Report, because Baumgart does not mention such "functions" in any form. As a further example, dependent claims 3, 44 & 85 recite limitations including "transforming zero or more function parameters" which the Examiner accepts are not disclosed by Adams, and which are also not disclosed by Baumgart. Here, the Examination Report refers to Baumgart at column 9, lines 2-20 with respect to the "binder program 24". However, Baumgart does not teach or suggest invoking a function or transforming function parameters as recited in these dependent claims.

The applicant is encouraged to note that dependent claims 19-28, 60-69 and 101-110 would be allowable if rewritten in independent form. However, for the reasons given above, we submit that the original independent claims 1, 42 & 83 are allowable with the clarifying amendments requested herein.

Filed herewith is a Request for a Three-Month Extension of Time, which extends the statutory period for response to expire on August 2, 2007. Accordingly, Applicant respectfully submits that this response is being timely filed.

US1DOCS 6187143v1

In view of the above amendment, applicant believes the pending application is in condition for allowance. No other fees are believed to be due in connection with the filing of this response, however the Commissioner is authorized to debit Deposit Account No. 08-0219 for any required fee necessary to maintain the pendency of this application.

Respectfully submitted,

Dated: July 10, 2007

Ronald R. Demsher
Registration No.: 42,478
Attorney for Applicant(s)

Wilmer Cutler Pickering Hale and Dorr LLP
60 State Street
Boston, Massachusetts 02109
(617) 526-6000 (telephone)
(617) 526-5000 (facsimile)

32